

In the Claims:

1. (Previously presented) A method of assigning a predetermined maximum number of logical ports to respective peripheral devices physically connected to a server in a network having client computers with client software, wherein the respective peripheral devices are connected to one or more physical port interfaces, the assigned logical ports enabling client software to communicate with the peripheral devices regardless of the particular physical port interface the peripheral device is connected to, the method comprising the steps of:

the server querying, responsive to a predetermined event, a peripheral device that is connected to the server by a physical port interface for information which uniquely identifies the peripheral device, wherein said peripheral device identification information comprises at least the manufacturer's serial number of the peripheral device;

the peripheral device communicating said identification information to the server;

the server initially assigning a logical port identification of an available unassigned one of the logical ports for a peripheral device that is connected to a physical port interface of the server and storing said logical port identification and said unique identification information in memory;

the server thereafter directing communications for respective peripheral devices from client computers to said assigned logical port for such peripheral devices, wherein the communications are executed utilizing the client software.

2. (Original) A method as defined in claim 1 wherein the peripheral devices are connected to the server via a physical parallel port interface.

3. (Original) A method as defined in claim 1 wherein the peripheral devices are connected to the server via a physical Universal Serial Bus interface.

4. (Original) A method as defined in claim 1 wherein said predetermined event is the initialization of the server.

5. (Original) A method as defined in claim 1 wherein said predetermined event is the connection of a previously unconnected peripheral device to the server.

6. Cancelled

7. (Previously presented) A method as defined in claim 1 wherein said logical port identification and said unique identification information are stored in a table having a predetermined maximum number of logical port entries.

8. (Original) A method as defined in claim 7 wherein the status of each of said maximum logical port entries is maintained in said table, with logical ports that are assigned having an assigned status and logical ports that are unassigned having a free status.

9. (Previously presented) A method as defined in claim 8 wherein when the server, responsive to said predetermined event, queries peripheral devices that are then connected to the server, compares said unique identification information received therefrom with said information maintained in said table and utilizes said assigned logical ports for directing communications to the connected peripheral devices in response to a positive comparison.

10. (Previously presented) A method as defined in claim 8 wherein when the server, responsive to said predetermined event, queries peripheral devices that are then connected to the server, compares said unique identification information received therefrom with said information maintained in said table and fails to detect a positive comparison, performs the following steps with respect to each peripheral device for which a positive comparison is not found:

the server searches the table to determine if the number of logical port entries is less than the maximum number of entries, indicating that one or more logical port entries are available;

the server assigns an available logical port identification to an unassigned peripheral device and stores said logical port identification, said identification information and an assigned status in an entry in the table in response to a logical port being available;

the server suspends processing of the unassigned peripheral device for a predetermined time period if there is no logical port available;

the server resumes processing after said predetermined time period and searches said table for reserved status entries and selects a reserved entry that closely matches, according to predetermined criteria, the unique identification information of

the peripheral device that is unassigned and assigns the unassigned peripheral device to the logical port.

11. (Original) A method as defined in claim 10 wherein unassigned peripheral devices are placed in a queue for subsequent processing.

12. (Original) A method as defined in claim 1 wherein said memory is a non-volatile memory.

13. (Previously presented) A method of assigning a predetermined maximum number of logical ports to respective peripheral devices physically connected to a server in a network having client computers with client software, wherein the respective peripheral devices are adapted to be connected to one or more physical port interfaces, the assigned logical ports enabling client software to communicate with the peripheral devices regardless of the particular physical port interface the peripheral device is connected to, the method comprising the steps of:

the server querying, responsive to a predetermined event, a peripheral device that is connected to the server by a physical port interface for information which uniquely identifies the peripheral device, wherein said peripheral device identification information comprises at least the manufacturer's serial number of the peripheral device;

the peripheral device communicating said unique identification information to the server;

the server placing the unassigned peripheral devices in a queue for subsequent assignment of unassigned logical ports by manual assignment;

the server storing assigned logical port identification and said unique identification information in memory;

the server thereafter directing communications for respective peripheral devices from client computers to said assigned logical port for such peripheral devices wherein said communications are executed utilizing the client software.

14. (Previously presented) A method as defined in claim 13 wherein said logical port identification and said unique identification information are stored in a table having said predetermined maximum number of logical port entries, said logical

ports that are assigned having an assigned status and logical ports that are unassigned having a free status.

15. (Previously presented) A system of assigning a predetermined maximum number of logical ports to respective peripheral devices physically connected to a server connected in a network including client computers having client software for controlling the operation of the peripheral devices, wherein the respective peripheral devices are connected to one or more physical port interfaces, the assigned logical ports enabling client software to communicate with the peripheral devices regardless of the particular physical port interface the peripheral device is connected to, the system comprising:

the server being adapted to query, responsive to a predetermined event, a peripheral device that is connected to the server by a physical port interface for information which uniquely identifies the peripheral device, wherein said peripheral device unique identification information comprises at least the manufacturer's serial number of the peripheral device;

the peripheral device being adapted to communicate said identification information to the server;

the server being adapted to initially assign a logical port identification of an available logical port for a peripheral device that is connected to a physical port interface of the server and store said logical port identification and said unique identification information in memory;

the server thereafter directing communications for respective peripheral devices from client computers to said assigned logical port for such peripheral devices, wherein the client computers communicate with the peripheral device using the client software.

16. (Previously presented) A system as defined in claim 15 wherein the server stores logical port identification and said unique identification information in a table having said predetermined maximum number of logical port entries.

17. (Original) A system as defined in claim 16 wherein the server stores the status of each of said maximum logical port entries in said table, with logical ports

that are assigned having an assigned status and logical ports that are unassigned having a free status.

18. (Previously presented) A system as defined in claim 17 wherein when the server, responsive to said predetermined event, queries peripheral devices that are then connected to the server, compares said unique identification information received therefrom with said information maintained in said table and directs communications to the connected peripheral devices using said assigned logical ports in response to a positive comparison.

19. (Previously presented) A system as defined in claim 18 wherein when the server, responsive to said predetermined event, queries peripheral devices that are then connected to the server, compares said unique identification information received therefrom with said information maintained in said table and fails to detect a positive comparison, the server performs the following steps with respect to each peripheral device for which a positive comparison is not found:

the server searches the table to determine if the number of logical port entries is less than the maximum number of entries, indicating that one or more logical port entries are available;

the server assigns an available logical port identification to an unassigned peripheral device and stores said logical port identification, said identification information and an assigned status in an entry in the table in response to a logical port being available;

the server suspends processing of the unassigned peripheral device for a predetermined time period if there is no logical port available;

the server resumes processing after said predetermined time period and searches said table for reserved status entries and selects a reserved entry that closely matches, according to predetermined criteria, the identification information of the peripheral device that is unassigned and assigns the unassigned peripheral device to the logical port.

20. (Previously presented) A computer program product comprising a computer usable medium having computer readable program codes embodied in the medium that when executed causes the server to:

query a peripheral device that is connected to the server by a physical port interface for information which uniquely identifies the peripheral device, responsive to a predetermined event, wherein said unique peripheral device identification information comprises at least the manufacturer's serial number of the peripheral device;

receive identification information that is transmitted by the peripheral device;

assign a logical port identification for the peripheral device that transmitted the unique identification information and store said logical port identification and said unique identification information in memory;

thereafter direct communications for the peripheral device from client computers to said assigned logical port for such peripheral devices, wherein the communications are executed utilizing the client software.

21. (Original) A product as defined in claim 20 which causes the server to store the status of each of said logical port entries in said table, with logical ports that are assigned having an assigned status and logical ports that are unassigned having a free status.

22. (Original) A product as defined in claim 21 which causes the server, when responsive to said predetermined event, to query peripheral devices that are then connected to the server, to compare said identification information received therefrom with said information maintained in said table and direct communications to the connected peripheral devices using said assigned logical ports in response to a positive comparison.

23. (Previously presented) A product as defined in claim 22 which causes the server, when responsive to said predetermined event, to query peripheral devices that are then connected to the server, to compares said unique identification information received therefrom with said information maintained in said table and fails to detect a positive comparison, perform the following steps with respect to each peripheral device for which a positive comparison is not found:

search the table to determine if the number of logical port entries is less than a maximum number of entries, indicating that one or more logical port entries are available;

assign an available logical port identification to an unassigned peripheral device and store said logical port identification, said unique identification information and an assigned status in an entry in the table in response to a logical port being available;

suspend processing of the unassigned peripheral device for a predetermined time period if there is no logical port available;

resume processing after said predetermined time period and search said table for reserved status entries and select a reserved entry that closely matches, according to predetermined criteria, the unique identification information of the peripheral device that is unassigned and assign the unassigned peripheral device to the logical port.

24. (Previously presented) A product as defined in claim 23 where said identification of information that approximates said unique identification information comprises at least identical manufacturer and model information.